Remarks

I. Background and Summary

This application was filed September 13, 2001. A non-final rejection was mailed December 17, 2004, rejecting claims 1-4, 6-24 and 28-30 on §102 grounds. Applicants responded to that rejection on March 2, 2005, primarily by pointing out that the §102 rejection was incorrect. The Examiner issued a Final Rejection June 3, 2005, essentially reiterating his earlier rejection. An interview between the Examiner and applicants' attorney was conducted, and a proposed amendment was discussed, without achieving a consensus. An Appeal Brief was filed November 10, 2005.

The latest Office Action withdraws the case from Appeal and reopens prosecution with essentially the same §102 rejections that were grounds for the Appeal, and also presents new §103 rejections based on a new reference. Applicants respectfully assert that the Examiner has implicitly conceded that the §102 rejections were improper by withdrawing the case from Appeal rather than having those rejections scrutinized, and therefore those rejections will receive minimal treatment at this time. In an effort to expedite the already extended prosecution of this case, however, applicants have amended certain claims to recite a "DMA command complete queue," which is clearly not taught or suggested in the cited art. As noted in the Abstract of the present invention, an advantage of this limitation is that "The processor need not monitor multiple DMA commands to make sure they have all been completed before the software branch is taken, but rather the processor pops a DMA command complete queue to make sure that the last of the DMA commands has been completed."

II. §102 Rejections

Claims 1-4, 6-24 and 28-34 stand rejected under 35 U.S.C. §102 over what the Examiner calls "Applicant Admitted Prior Art (AAPA)."

The Office Action states, on page 2, lines 14-22:

> Fig. 2, which features a QUEUE MNGR 209 with elements 218 (labeled DMA CMD COMPLETE QUEUE) and 217 (labeled DMA CMD QUEUE) differs from Fig. 1, which is labeled as "PRIOR ART" featuring the same QUEUE MNGR 108 with the same elements corresponding to

> 217 and 218 (non-labeled). In other words, Fig. 2, representing the claimed invention differs from Fig. 1 (prior art) by simply labeling elements 217 and 218. Nevertheless, Applicant contends that the recited "DMA command queue" differs from "DMA commands" of Applicant Admitted Prior Art (AAPA) of Figure 1 by the simple fact that the Prior Art does not label the DMA commands, completed or not.

As noted above, applicants respectfully assert that the Examiner has conceded that this rejection is improper by withdrawing the case from Appeal rather than having the rejection scrutinized. Briefly, however, FIG. 2 differs from FIG. 1 (Prior Art) in several important ways. First, FIG. 1 features a DMA command register 116 and a DMA command complete register 117, each register coupled to processor 109 and to DMA controller 115. FIG. 2 does not feature either a DMA command register or a DMA command complete register. Second, while FIG. 1 features two unlabeled elements within QUEUE MNGR 108, these unlabeled elements are different from the DMA CMD QUEUE 217 and the DMA CMD COMPLETE QUEUE 218 of FIG. 2, and these unlabeled elements in FIG. 1 perform different functions from queues 217 and 218 of FIG. 2. Prior art QUEUE MNGR 108 manages multiple queues serving varied purposes, but none of those queues is a DMA command queue or a DMA command complete queue.

The Office Action further states, on page 3, lines 3-8:

> Moreover, the examiner would like to emphasize that an out-of-order sequencing process defines sequence process with no particular order, different from a known or particular order sequencing process, such as FIFO or LIFO; but an out-of-order sequence is a process with no particular order, which is also known way for ORDERING processes without a fixed or particular order. Therefore an out-of-order is also an order in command processing.

It is unclear which claim or claim element, if any, the preceding statement pertains to, and applicants await such information before responding to the statement. It should be noted however, that the Examiner's reasoning in this regard removes all meaning from the word "order," as he states that "an out-of-order sequence is a process with no particular order, which is also known way for ORDERING processes without a fixed or particular order." Although claims may be construed in as broad a manner as is

reasonable, applicants respectfully assert that this line of reasoning that would remove all meaning from words is so broad as to be absurd.

The Office Action further states, on page 3, lines 9-12:

> Computer Science Dictionary Defines queue as:
>
> a. A sequence of stored data or programs awaiting processing.
>
> b. A data structure from which the first item that can be retrieved is the one stored earliest.

Applicants respectfully object to this reference to an unknown "Computer Science Dictionary," and respectfully request the Examiner to provide a copy of the reference as well as a citation. Moreover, as noted in *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998)

> a common meaning, such as one expressed in a relevant dictionary, that flies in the face of the patent disclosure is undeserving of fealty. As one of our predecessor courts stated in *Liebscher v. Boothroyd*, 46 C.C.P.A. 701, 258 F.2d 948 (CCPA 1958):
>
> Indiscriminate reliance on definitions found in dictionaries can often produce absurd results. . . . One need not arbitrarily pick and choose from the various accepted definitions of a word to decide which meaning was intended as the word is used in a given claim. The subject matter, the context, etc., will more often than not lead to the correct conclusion.

Assuming arguendo that the Examiner's selected dictionary definition is correct, the §102 rejection is still misplaced, as discussed in the Appeal Brief, which the Examiner chose not to contest.

III. §103 Rejections

Claims 1-4, 6-24 and 28-34 stand rejected under 35 U.S.C. §103(a) as being unpatentable over "Applicant Admitted Prior Art (AAPA)" in view of U.S. Patent No. 5,745,781 to Ekanadham et al. (hereinafter "Ekanadham").

Independent claim 1; dependent claims 2-4, 6-10

Initially, applicants note that page 5, line 17 – page 6, line 12 of the latest Office Action are merely a reprint of the §102 rejection on page 6, line 10 – page 7, line 4 of the Final Rejection which, as noted above, has been conceded as improper by the Examiner

by withdrawing the Appeal rather than facing scrutiny. Because the argument on page 5, line 17 – page 6, line 12 of the latest Office Action is not effective in a §102 rejection, and because the latest Office Action provides no reason why it would be modified by one of ordinary skill in the art to teach what is claimed, that argument is also defective in a §103 rejection. For at least this reason, the Office Action has not provided a *prima facie* case of obviousness for claim 1 or any claim that depends from claim 1.

To wit, the Office Action asserts that "AAPA" teaches "a DMA command queue [NID 100 with command queue 122 in SRAM 112]" (Office Action page 5, line 22). Applicants respectfully submit that "AAPA" does not teach a "DMA command queue 122". Rather, "AAPA" teaches "DMA commands 122 stored in SRAM 112" (Par. 0005). "DMA commands" are not the same thing as a "DMA command queue"; this point has been made in several previous communications by Applicants, including the Appeal Brief.

The Office Action further states, on page 6, lines 12-20:

> AAPA fail to clearly teach the claimed DMA command with the network interface card. However, Ekanadham discloses communication adapter 400, Fig. 2a (network interface device) interfacing multiple devices in a network including a DAM engine 482, a DAM command queue 480 in which DMA commands are placed and outgoing queue 484 for storing outgoing packets [Col. 6, Line 60 to Col. 7, Line 25]. Therefore, it would have been obvious to one of ordinary skill in the art to include the DMA command queue of Ekanadham within the network interface device of the Prior Art as queues are used to correlate requests generated by one or more client processes to a service process.

Applicants assume that "the claimed DMA command" in the preceding statement in fact refers to the claimed DMA command queue, and that the absence of the word "queue" is a typographical error, not an attempt to ignore the queue limitation. Claim 1 has been amended to recite a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue. Applicants respectfully assert that Ekanadham does not teach a network interface device including a DMA command queue and a DMA command complete queue, in contrast to claim 1. The outgoing packet queue 484 of Ekanadham stores outgoing data packets; it is an outgoing packet queue, not a DMA command complete queue. Claim 1 as currently amended recites a network interface

device including a DMA command queue and a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue. For at least this reason, claim 1 is nonobvious over the cited art.

With respect to claim 2, the Office Action asserts that "AAPA" teaches "a DMA command complete queue on the network interface device [DMA command complete queue 17]" (Office Action page 7, lines 2-3). Applicants respectfully submit that "AAPA" does not teach a "DMA command complete queue 17". Rather, "AAPA" teaches a "DMA command complete register 117" (Par. 0003). A register is not the same thing as a queue; this point has been made in several previous communications by Applicants, including the Appeal Brief. The Office Action asserts that "Ekanadham further discloses maintaining a DMA command outgoing queue 484 on the network interface device 400, the DMA controller pushing values onto the DMA command outgoing queue 480" (Office Action page 7, lines 5-7). Applicants respectfully submit that the outgoing packet queue 484 of Ekanadham stores outgoing data packets; it is an outgoing packet queue, not a DMA command complete queue. The language of claim 2 has been incorporated into claim 1, and claim 2 has been canceled.

Claim 3 has been amended to depend from claim 1 instead of claim 2. The Office Action asserts that "AAPA" teaches that "the processor uses the DMA command complete queue to determine that the first and second portions of data are both present in local memory." Applicants respectfully submit that "AAPA" does not teach a "DMA command complete queue". Rather, "AAPA" teaches a "DMA command complete register 117" (Par. 0003). A register is not the same thing as a queue; this point has been made in several previous communications by Applicants, including the Appeal Brief.

Claim 4 has been amended to recite that the queue manager hardware maintains the DMA command queue and the DMA command complete queue in SRAM. The Office Action asserts that "AAPA" teaches "the queue manager hardware maintaining the DMA command queue" in SRAM (Office Action page 7, lines 17-18). Applicants respectfully assert that "AAPA" does not teach a DMA command queue in SRAM. Rather, "AAPA" teaches "DMA commands 122 stored in SRAM 112". "DMA

commands" are not the same thing as a DMA command queue; this point has been made in several previous communications by Applicants, including the Appeal Brief.

Claims 6-10 depend from claim 1, and are consequently allowable for at least the same reasons that claim 1 is allowable.

Independent claim 11; dependent claims 12-13

Initially, applicants note that page 8, line 10 – page 9, line 7 of the latest Office Action are merely a reprint of the §102 rejection on page 6, line 10 – page 7, line 4 of the Final Rejection which, as noted above, has been conceded as improper by the Examiner by withdrawing the Appeal rather than facing scrutiny. Because the argument on page 8, line 10 – page 9, line 7 of the latest Office Action is not effective in a §102 rejection, and because the latest Office Action provides no reason why it would be modified by one of ordinary skill in the art to teach what is claimed, that argument is also defective in a §103 rejection. For at least this reason, the Office Action has not provided a *prima facie* case of obviousness for claim 11 or any claim that depends from claim 11.

To wit, the Office Action asserts that "AAPA" teaches "a DMA command queue [DMA command queue 122 in SRAM 112]" (Office Action page 8, lines 14-15). Applicants respectfully submit that "AAPA" does not teach a "DMA command queue 122". Rather, "AAPA" teaches "DMA commands 122 stored in SRAM 112" (Par. 0005). "DMA commands" are not the same thing as a "DMA command queue"; this point has been made in several previous communications by Applicants, including the Appeal Brief.

The Office Action states, on page 9, lines 7-15:

> AAPA fail to clearly teach the claimed DMA command with the network interface card. However, Ekanadham discloses communication adapter 400, Fig. 2a (network interface device) interfacing multiple devices in a network including a DAM engine 482, a DAM command queue 480 in which DMA commands are placed and outgoing queue 484 for storing outgoing packets [Col. 6, Line 60 to Col. 7, Line 25]. Therefore, it would have been obvious to one of ordinary skill in the art to include the DMA command queue of Ekanadham within the network interface device of the Prior Art as queues are used to correlate requests generated by one or more client processes to a service process.

Applicants assume that "the claimed DMA command" in the preceding statement in fact refers to the claimed DMA command queue, and that the absence of the word "queue" is a typographical error, not an attempt to ignore the queue limitation. Claim 11 has been amended to incorporate language similar to that of claim 2. Claim 11 now recites a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue.

Applicants respectfully assert that Ekanadham does not teach a network interface device including a DMA command queue and a DMA command complete queue. The outgoing packet queue 484 of Ekanadham stores outgoing data packets; it is an outgoing packet queue, not a DMA command complete queue. Claim 11 as amended recites a network interface device including a DMA command queue and a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue.

Claim 13 has been amended so that the queue manager hardware stores at least part of both the DMA command queue and the DMA command complete queue in SRAM. Claim 12 depends from claim 11, and is consequently allowable for at least the same reasons that claim 11 is allowable.

Independent claim 14; dependent claims 15-20

The Office Action does not provide any reason for the rejection of claim 14, and so does not present a *prima facie* case of obviousness for claim 14 or any claim that depends from claim 14. Moreover, claim 14 has been amended to recite a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue. Applicants respectfully submit that the Ekanadham patent cited with respect to several other claims does not teach a network interface device including a DMA command queue and a DMA command complete queue.

Claims 15-20 depend from claim 14, and are consequently allowable for at least the same reasons that claim 14 is allowable.

Independent claim 21; dependent claims 22-24

Initially, applicants note that page 4, lines 1-15 of the latest Office Action are merely a reprint of the §102 rejection on page 5 of the Final Rejection which, as noted above, has been conceded as improper by the Examiner by withdrawing the Appeal rather than facing scrutiny. Because the argument on page 4, lines 1-15 of the latest Office Action is not effective in a §102 rejection, and because the latest Office Action provides no reason why it would be modified by one of ordinary skill in the art to teach what is claimed, that argument is also defective in a §103 rejection. For at least this reason, the Office Action has not provided a *prima facie* case of obviousness for claim 21 or any claim that depends from claim 21.

To wit, the Office Action asserts that "AAPA" teaches "a DMA command queue [DMA commands 122]" (Office Action page 4, lines 4-5). Applicants respectfully submit that "AAPA" does not teach a "DMA command queue". Rather, "AAPA" teaches "DMA commands 122 stored in SRAM 112" (Par. 0005). "DMA commands" are not the same thing as a "DMA command queue"; this point has been made in several previous communications by Applicants, including the Appeal Brief.

The Office Action states, on page 9, lines 7-15:

> AAPA fail to clearly teach the claimed DMA command with the network interface card. However, Ekanadham discloses communication adapter 400, Fig. 2a (network interface device) interfacing multiple devices in a network including a DAM engine 482, a DAM command queue 480 in which DMA commands are placed and outgoing queue 484 for storing outgoing packets [Col. 6, Line 60 to Col. 7, Line 25]. Therefore, it would have been obvious to one of ordinary skill in the art to include the DMA command queue of Ekanadham within the network interface device of the Prior Art as queues are used to correlate requests generated by one or more client processes to a service process.

Applicants assume that "the claimed DMA command" in the preceding statement in fact refers to the claimed DMA command queue, and that the absence of the word "queue" is a typographical error. Claim 21 has been amended to incorporate language similar to that of claim 2. Claim 21 now recites a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue.

Applicants respectfully submit that Ekanadham does not teach a network interface device including a DMA command queue and a DMA command complete queue. The outgoing packet queue 484 of Ekanadham stores outgoing data packets; it is an outgoing packet queue, not a DMA command complete queue. Claim 21 as currently amended recites a network interface device including a DMA command queue and a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue.

Claims 22-24 depend from claim 21, and are consequently allowable for at least the same reasons that claim 21 is allowable.

Independent claim 28; dependent claims 29-30

The Office Action discusses claim 28 together with claim 21. Claim 28 has been amended to recite a DMA command complete queue, the DMA controller pushing values onto the DMA command complete queue, the processor popping values off the DMA command complete queue. Applicants' response regarding claim 28 is the same as applicants' response regarding claim 21; see text above regarding independent claim 21. For the reasons discussed above with regard to claim 21, the Office Action has not provided a *prima facie* case of obviousness for claim 28 or any claim that depends from claim 28.

Claims 29-30 depend from claim 28, and are consequently allowable for at least the same reasons that claim 28 is allowable.

Independent claim 31; dependent claims 32-34

Claim 31 has been amended to incorporate language similar to that of claim 2. Claim 31 now recites a DMA command complete queue within a network interface device, the DMA controller pushing values onto the DMA command complete queue, wherein the network interface device performs fast-path transport and network layer protocol processing.

The Office Action discusses claim 31 together with claim 21. Applicants' response regarding claim 28 includes applicants' response regarding claim 21; see text above regarding Independent claim 21. Moreover, applicants note that the invention of

claim 31 recites a network interface device that performs "fast-path transport," a limitation not discussed in the Office Action. For at least these reasons, the Office Action has not provided a *prima facie* case of obviousness for claim 31 or any claim that depends from claim 31.

Claims 32-33 depend from claim 31, and are consequently allowable for at least the same reasons that claim 31 is allowable. Claim 34 has been amended in two ways. Claim 34 now depends from claim 32. In the invention of claim 34, the DMA command complete queue is used to determine that the at least a part of one of the frames and the at least a part of another of the frames are both present in the second memory.

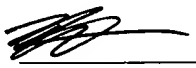
III. Allowed Claims

Applicants appreciate the allowance of claims 5 and 25-27.

IV. Conclusion

In this Amendment, applicants have demonstrated that the Office Action does not present a *prima facie* case of anticipation or obviousness for any of the pending claims. As such, applicants respectfully assert that the application is in condition for allowance, and a Notice of Allowance is solicited.

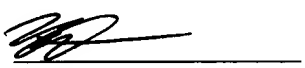
Respectfully submitted,



Mark Lauer
Reg. No. 36,578
6601 Koll Center Parkway
Suite 245
Pleasanton, CA 94566
Tel: (925) 484-9295
Fax: (925) 484-9291

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on April 27, 2006.

Date: 4-27-06 

Mark Lauer